

Мікроконтролер з системою команд, що розширюється

Сергієнко Анатолій Михайлович
 провідний інженер
 КПІ ім. Ігоря Сікорського
 Україна

Молчанов Олексій Андрійович
 аспірант
 КПІ ім. Ігоря Сікорського
 Україна

Запропонована архітектура восьмирозрядного стекового процесора SM8, який призначений для реалізації у програмованих логічних інтегральних схемах (ПЛІС). Мікроконтролер з цією архітектурою має невеликі апаратні витрати, скорочений об'єм програмного забезпечення а також можливість додавати користувачем до сотні нових команд до його системи команд.

Ключові слова: стековий процесор, Forth, ПЛІС, VHDL.

ВСТУП

При розробці системи на кристалі програмованої логічної інтегральної схеми (ПЛІС) часто виникає потреба організувати обмін даними через такі інтерфейси, як I2C, SPI, Ethernet та інші. Для цього часто використовують RISC-процесори, які керовані операційною системою з відповідним стеком протоколів обміну. Але у багатьох випадках обмін даними виконується за нескладним протоколом і з порівняно невеликою швидкістю, як наприклад, через інтерфейс I2C. При цьому раціональніше використати ядро мікроконтролера, яке має як невеликі апаратні витрати, так і прості процедури програмування та відлагодження. Крім того, такий мікроконтролер може замінити собою керуючий автомат.

Але для реалізації у ПЛІС важливо мати мінімізований об'єм матзабезпечення цього мікроконтролера, оскільки об'єм блоків пам'яті, вбудованих в ПЛІС, є суттєво обмеженим. При цьому бажано мати такий мікроконтролер, систему команд якого програміст власноруч може скоректувати під потреби проекту, для покращення зручності програмування, а також для мінімізації довжини програм.

МІКРОКОНТРОЛЕРИ ДЛЯ РЕАЛІЗАЦІЇ У ПЛІС

Найпоширенішою восьмирозрядною архітектурою мікроконтролера вважається архітектура i8051, яка реалізована у багатьох мікроконтролерних ядрах для ПЛІС. У роботі [1] досліджено особливості проектування його клона FS8051 для ПЛІС. Але навіть з відкиданням периферійних пристроїв та скороченням системи команд апаратні витрати залишаються доволі великими (см. Табл. 2).

Фірмою Xilinx для вказаних потреб пропонується поширене ядро мікроконтролера PicoBlaze. Воно описане на рівні логічних таблиць (LUT) та тригерів ПЛІС цієї фірми, як наприклад, ядро мікроконтролера KCP5M6 [2]. Тому воно має мінімальні апаратні витрати та доволі високу тактову частоту (см. Табл. 2). Хоча цей мікроконтролер обробляє восьмирозрядні дані, він керується 18-розрядними командами, кожна з яких виконується за два такти. Отже, для програмування мікроконтролерів цієї архітектури необхідні програми відносно великого об'єму. Так само, як і для архітектури i8051, для цієї архітектури неможливе додання нових команд.

Система команд стекового процесора відрізняється тим, що операнди мають неявну адресацію, тому що вони, як правило, розміщуються у небагатьох фіксованих регістрах стекової пам'яті. Тому такі команди мають короткий формат команд. Оскільки ці команди підтримують алгоритми, що активно використовують стекову адресацію, програми, які складені для такого процесора мають мінімізовану довжину [3].

Різними авторами розроблено декілька проектів стекових процесорів, які реалізовані в ПЛІС і які є доступними для відтворення [4-6]. Усі вони мають 16-розрядні команди та обробляють 16-розрядні дані. В роботі [6] показано, що стековий процесор має приблизно у 2,5 рази меншу довжину програми, ніж програма для процесора Xilinx MicroBlaze при реалізації протоколів обміну даними через інтерфейси. Крім того, усі стекові процесори дають змогу збільшити систему команд на кілька команд користувача. При цьому слід внести відповідні зміни до опису процесора на рівні регістрових передач, що часто є незручним для користувача процесора.

Отже, архітектура стекового процесора дає змогу мінімізувати об'єм вбудованого матзабезпечення та одержати мінімізовані апаратні витрати за рахунок реалізації спрощених команд. Крім того, для такої архітектури нескладно розробляти компілятори, тому що, як правило, система команд є підмножиною команд мови Forth. Відомо, що ця мова є зручною для реалізації як граматичного розбору рядків, так і для інтерпретації операторів мов високого рівня. Програма на мові асемблера стекового процесора має такий самий синтаксис, як програма на Forth [3]. Тому доцільно розробити таку архітектуру стекового процесора, яка дає не тільки мінімізовані апаратні витрати, але й спрощену реалізацію команд користувача.

МІКРОКОНТРОЛЕР SM8

Структура розробленого мікроконтролера SM8 показана на рисунку. Цей процесор має поширену двохстекову архітектуру. Він складається з лічильника команд PC, блоку пам'яті даних Data RAM, блоку пам'яті програм Program ROM, регістра команди IR, шифратора команд користувача CDC, стека адрес повернення RStack глибиною 33,

стека даних DStack глибиною 34, арифметико-логічний пристрій ALU та периферійні регістри R0,..., Rn, n < 32. Регістри T, N, P – це крайні регістри стеку Dstack і призначені для зберігання операндів та результату ALU. Program ROM об’ємом до 7936 байт разом з Data RAM об’ємом до 256 байт мають один адресний простір.

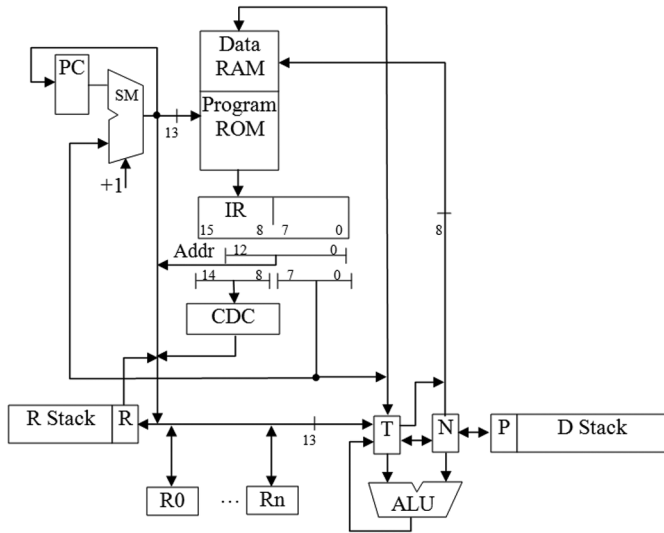


Рис. 1 Структура мікроконтролера SM8

У таблиці 1 представлена система команд мікроконтролера SM8. Усі команди, крім команд CALL, LIT та IF, є однобайтовими. Команди переходу та вводу константи виконуються за два такти, а решта команд – за один. Команди виконуються у конвеєрному режимі. Типові програми на Forth мають багато викликів підпрограм, завдяки чому вони є короткими. За рахунок цього, а також через часте використання команд LIT та IF, середня тривалість виконання однієї команди в даній архітектурі складає півтора такти.

Команди користувача реалізуються наступним чином. Після запису такої команди у регістр IR її код перекодується в CDC у адресу підпрограми. Ця підпрограма викликається з бібліотеки підпрограм команд користувача і виконує усі дії, які необхідні для реалізації даної команди. Перехід до наступної команди програми виконується за командою RET. У цій підпрограмі можна також читати та обробляти поля операндів, які слідує за байтом коду операції. При цьому належним чином коректується адреса повернення у регістрі R.

ТАБЛИЦЯ 1. СИСТЕМА КОМАНД МІКРОКОНТРОЛЕРА SM8

Ім'я	Формат	Дія
CALL	001 Addr	PC+1 -> R, PC = Addr, виклик підпрограми
INR	010 n	Rn -> T, прийом даного
OUTR	011 n	Rn = T, вивід даного
NOP	0000 0000	Немає операції
LIT	0000 0001 B	B -> T, ввід константи
IF	0000 0010 D	PC = PC + D при T = 0, інакше PC = PC + 1
DUP	0000 0100	N = T->
OVER	0000 0101	P = T->
SWAP	0000 1001	X = T, T = N, N = X
@	0000 1010	T = RAM[T], читання пам'яті,
!	0000 1011	DRAM[T] = N, запис в пам'ять, T->
R>	0000 1100	R -> T
>R	0000 1110	T -> R
RET	0000 1101	R -> PC, повернення з підпрограми

DROP	0000 1111	T->, опорожнення стеку
NOT	0001 0000	T = not T
OR	0001 0001	T = T or N
AND	0001 0010	T = T and N
XOR	0001 0011	T = T xor N
ADD	0001 1000	T = T + N
INC	0001 1001	T = T + 1
SUB	0001 1010	T = T - N
DEC	0001 1011	T = T - 1
	1xxx xxxx	Команда користувача

Команди користувача можуть зберігатись як в Program ROM, так і в Data RAM. Таким чином, в мікроконтролер можна записувати певний динамічний сценарій обробки даних, який записано командами користувача або виконувати розбір рядка, який набраний з таких команд. Наприклад, це може бути рядок з команд десяткового калькулятора.

У таблиці 2 представлені результати синтезу мікроконтролера SM8 у ПЛІС Xilinx Spartan-6 при встановленні параметрів оптимізації за апаратними витратами. Також для порівняння представлені параметри перелічених вище мікроконтролерів, які були синтезовані у таких самих умовах. Аналіз таблиці показує, що мікроконтролер SM8 має найнижчі апаратні витрати у числі LUT і конфігурованих логічних блоків (CLB) та найбільшу продуктивність у мільйонах команд за секунду (MIPS) серед стекових процесорів. Це роз'яснюється тим, що скорочення даних, що обробляються, до восьми розрядів, дало змогу зменшити як апаратні витрати, так і затримку в ALU.

ТАБЛИЦЯ 2. ПАРАМЕТРИ ЯДЕР МІКРОКОНТРОЛЕРІВ ПРИ ЇХ КОНФІГУРУВАННІ У ПЛІС XILINX SPARTAN-6

Мікро-контролер	Розрядність команд	Апаратні витрати		Макс. тактова частота, МГц	Продуктивність, MIPS
		LUT	CLB		
FS8051 [1]	8, 16, 24	1293	470	89	30
KCPSM6 [2]	18	87	26	140	70
MSL16 [4]	16	235	61	100	67
b16-small [5]	16	280	73	100	50
J1 [6]	16	342	93	106	70
SM8	8, 16	181	50	140	94

ВИСНОВКИ

Запропонований мікроконтролер SM8 має невеликі апаратні витрати при високій продуктивності та скороченому об'ємі програм і призначений для реалізації нескладних алгоритмів керування, наприклад, для керування обміном даних через інтерфейс I2C. Проект мікроконтролера описано мовою VHDL і він може бути реалізований у ПЛІС будь-якої серії. Програміст має можливість додавати власні команди до його системи команд без зміни його опису. Заплановано, крім асемблера, розробити для цього мікроконтролера компілятор з мови високого рівня для спрощення проектування пристроїв, які реалізують протоколи обміну через такі інтерфейси, як I2C, SPI, Ethernet.

ПЕРЕЛІК ПОСИЛАНЬ

1. Сергієнко А.М., Лепеха В.Л. Деякі особливості проектування мікроконтролерів для CHK // Вісник НТУУ «КПІ», сер. Інформатика, управління та обчислювальна техніка. – Т. 50. – 2009. – С. 70-73.

2. Chapman, K. PicoBlaze for Spartan-6, Virtex-6, and 7-Series (KCPSM6). – Xilinx, Inc. – 2012. —118 p.
3. Koopman P. Stack computers: the new wave. — CA: Ellis Horwood, Mountain View Press. – 1989. – 234 p.
4. Leong P. H. W., Tsang P.K., Lee T.K. A FPGA Based Forth Microprocessor // Proc. of the IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM). – USA, Napa Valley, California. – 1998. — P.1-2.
5. Paysan B. b16-small—Less is More // EuroForth 2004 Proceedings, Jul. 9, 2006, pp. 1-8.
6. Bowman J., Garage W. J1: a small Forth CPU Core for FPGAs // Proc. EuroForth'2010, January, 2010. – P. 1–4.